



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/728,017	12/04/2003	Eric Lawrence Barsness	ROC920030312US1	7311
30206	7590	02/13/2008	EXAMINER	
IBM CORPORATION			STEELMAN, MARY J	
ROCHESTER IP LAW DEPT. 917			ART UNIT	PAPER NUMBER
3605 HIGHWAY 52 NORTH			2191	
ROCHESTER, MN 55901-7829			MAIL DATE	DELIVERY MODE
			02/13/2008	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	10/728,017	BARSNESS ET AL.
	Examiner	Art Unit
	MARY STEELMAN	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1) Responsive to communication(s) filed on 08 December 2007.  
 2a) This action is **FINAL**.                            2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4) Claim(s) 1,2 and 4-20 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1,2,4-20 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on \_\_\_\_\_ is/ are: a) accepted or b) objected to by the Examiner.  
     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) Notice of References Cited (PTO-892)  
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3) Information Disclosure Statement(s) (PTO/SB/08)  
     Paper No(s)/Mail Date \_\_\_\_\_

4) Interview Summary (PTO-413)  
     Paper No(s)/Mail Date. \_\_\_\_\_

5) Notice of Informal Patent Application  
 6) Other: \_\_\_\_\_

**DETAILED ACTION**

1. This Office Action is in response to Amendments and Remarks received 12/08/2007.

Per Applicant's request, claims 1, 2, 8, 9, 13, and 17 are amended. Claim 3 is previously cancelled. Claims 1, 2, and 4-20 are pending.

***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 8-12 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 8 recites, "An apparatus comprising: means for..." The Specification, page 7, line 22, discloses that the means may be implemented using any suitable hardware and / or software... As such, a software embodiment is disclosed, which is a non-statutory embodiment of 'software per se.' Examiner maintains the rejection of claims 8-12.

The Specification provides for the means of the apparatus to be software, all the elements of the claimed Apparatus implemented in software alone. It appears to Examiner that the intent is to claim the statutory category as a Machine (apparatus). This rejection may be overcome by

amending claim language to include hardware, such as a processor, a storage device, as disclosed on page 3, line 24 of Specification. Such an amendment explicitly excludes the 'software alone' embodiment.

3. Claims 1-20 are rejected under 35 U.S.C. 101 because the disclosed invention is inoperative and therefore lacks utility.

As examiner understands the amended claim language, an editor, a 'copy and paste' (some embodiments a punch-out) operation, a selection of a block of code, and a 'paste' location are received. A created method comprising a copy of the block of code, that exceeds a size threshold, will cause an invocation/call to be pasted at the paste location (redirect code to another memory location). A created method comprising a copy of the block of code, that is less than or equal to a size threshold, will be pasted into the paste location. Note that there is no indication that the application removes the original block of code. The outcome will be an application, including the original block of code, and also including a paste location, that either holds a created method holding a copy of the block of code or an invocation/call to a created method comprising a copy of the block of code. How does this new application work? The block of code is executed once, twice? The original block of code was cut out or overwritten? In the case of exceeding the threshold, the created method is stored (?) (where?) and the invocation accesses the stored created method? Where is the paste location? Is the paste location the same as the location of the original block of code? Did the created method or the invocation overwrite the original block of code at the paste location?

If 'the operation' is a punch-out (cut operation), the outcome will be an application, with the selected block of code cut out. The selected block of code is incorporated into a created method and pasted at a paste location or an invocation/call, of the created method holding the selected block of code, is pasted at a paste location. Is the created method or the invocation call stored at the paste location? If the invocation call is stored at the paste location, then the created method holding the selected block of code is stored (?) at some other location in memory and accessed by the invocation call. Is the paste location where the punch-out operation removed the selected block of code?

It seems that Applicant intends to claim an instruction restructuring algorithm, placing selected code into a created method and inlining the created method (store method into application) or redirecting via an invocation call to the created method which is stored elsewhere.

For a given application, and a given block of code, remove a selected block of code from the application and create a method incorporating the block of code. The block of code is pulled out of the code sequence and replaced with a created method incorporating the block of code or replaced with an invocation call to the created method incorporating the block of code, said method stored elsewhere.

The created method, incorporating the block of code, is stored at the paste location (where the original code is cut out), if the size is smaller than or equal to a threshold.

OR

an invocation call is added at the paste location (where the original code is cut out) to redirect execution to the created method which is stored elsewhere (in a larger location).

Claim 1 language does not recite removing code from original location in any case. Other independent claims do not recite removing code from original location in the case where the block of code is less than or equal to the threshold. Claim 1 does not recite replacing (cutting) if it exceeds a threshold. Therefore, for the 'less than or equal to the threshold' case or for claim 1, the block of code exists twice (at the original location and in the created method).

Where is the paste location? Is paste location where 'the original block of code" is removed?

Claim language does not store the created method somewhere (at the paste location where the original block of code was cut, or in a larger location).

Claim language recites original code block is only 'replaced' (cut) (claim 1 does not recite replacing) if it exceeds a threshold. Therefore, for the 'less than or equal to the threshold' case (or for claim 1), the block of code will exist twice.

***112 Rejections***

4. In view of the amendments to claims and Applicant's comments, the prior second paragraph of 35 U.S.C. 112 rejections are hereby withdrawn.

***Claim Objections***

5. Claim 8 recites “means for creating a method that comprises the block of and replacing the block of code...”, should be --means for creating a method that comprises the block of code and replacing the block of code...” Add the word ‘code’.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1, 2, 4-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 7,110,936 B2 to Hiew et al., in view of 2002 / 0032804 A1 Hunt, and further in view of USPN 5,740,443 to Carini.

Per claim 1:

A method comprising:

receiving an operation, a selection of a block of code in an application, and a selection of a paste location in the application;

in response to the receiving, determining whether the block of code has a size greater than a threshold;

if the block of code has a size greater than the threshold, creating a method that comprises the block of code, and

adding a first invocation of the method at the paste location;

if the block of code does not have a size greater than the threshold, performing the operation.

Hiew: Col. 7: 3-22, editor 80 (receiving an operation), individual procedures and data blocks as active elements 82 (a selection of code), selecting active elements, to cut, copy or move code segment... Col. 8: 43-50, site manager 70 and document manager 60 interact to effectuate the transfer of code between the remote server computers 22a and the local computer 22b...modules could be combined or further divided to form one single module or additional modules Col. 10: 3-4, statistics - max Col. 14: 63.- col. 15: 1, The site manager 70, the document manager 60, and the editor 80 work in conjunction with one another to execute the code...site manager 70 manages the connections...document manager 60 tracks the associations between the code (invocations) Col. 15: 65-66, visualizer window 121 and represent the procedures (method block of code) and data blocks as program flow icons 126. Col. 16: 31-35, allows changes to the code to be made...by using the editor...

Hiew disclosed redirecting code, but failed to explicitly disclose adding a first invocation.

However, Hunt disclosed:

[0265] To apply inline redirection, the COIGN runtime, a collection of DLLs, is loaded into the application's address space before the application executes. One of these DLLs, the COIGN runtime executive (RTE), inserts the inline redirection code. [0263] Referring to FIG. 14, at load time, the first few instructions 502 of the target function 500 are replaced with a jump instruction 504 to the instrumentation detour function 506 in the instrumentation. The first few instructions 502 are normally part of the function prolog generated by a compiler and not the targets of any branches. The replaced instructions 502 are used to create a trampoline function 508. When the modified target function 501 is invoked, the jump instruction 504 transfers execution to the detour function 506 in the instrumentation. The detour function 506 passes control to the remainder of the target function by invoking the trampoline function 508. After the moved instructions 502 are executed in the trampoline 508, a jump instruction 510 transfers execution back to a spot in the target function 501. The trampoline function 508 allows the detour function 506 to invoke the target function without interception.

Hiew / Hunt failed to suggest copying and pasting is done in response to a condition, test regarding the size of the block of code. (if the block of code has a size greater than the threshold)

However, Carini disclosed inlining code according to a cost function (col. 10: 23-28). The size of the procedure and the size of the region are given consideration prior to inlining (pasting in a method comprising a block of code).

Therefore, it would have been obvious, to one of ordinary skill in the art, to modify Hiew, using the teachings of Hunt, because Hiew recognized the need for (col. 1: 64-65) an Integrated Development Environment for generating and maintaining software code. Likewise Hunt recognized the benefit of [0014-0015] modifying an application, packaging it differently to optimize distribution and execution. It would have been obvious to further modify Hiew / Hunt using the teachings of Carini because Carini was also directed towards compiler optimization and site specific inlining. One would be motivated to place code into a location that was large enough to fit the code or otherwise locate the code elsewhere.

Per claim 2.

-replacing the block of code at a copy location in the application with a second invocation of the method.

Hiew: Col. 8: 43-50, site manager 70 and document manager 60 all preferably interact with one another to effectuate the transfer of code between the remote server computers 22a and the local computer 22b. It should also be understood that each of these modules could be combined or further divided to form one single module or additional modules.

Per claim 4:

-displaying user interface options based on whether the operation is to be performed within a single part and class.

Hiew: Col. 10: 22-28, activation of the integrate software application 30...the graphical user interface 40 will be displayed and the user will have the opportunity to open modules...or engage functions represented by buttons (user interface options)... Col. 8: 43-50, site manager 70 and document manager 60 all preferably interact with one another to effectuate the transfer of code between the remote server computers 22a and the local computer 22b. It should also be understood that each of these modules could be combined or further divided to form one single module or additional modules (within a single part and class).

Per claim 5:

-displaying user interface options based on whether the operation is to be performed between two parts in a single package.

Hiew: Fig. 11 & Col. 10: 33-43, displays the defined / available remote computers 22a...displays a server profile for a selected remote computer 22a, comprising connection information... Col. 11: 49-50, The user may then select one or more of the available files for editing or visualizing.

Per claim 6:

-the operation comprises a copy-and-paste operation.

Hiew: Col. 11: 61, copying...pasting information

Per claim 7:

-the operation comprises a punch-out operation.

Hiew: Col. 11: 61, cutting (punch-out)...

Per claim 8:

An apparatus comprising:

means for receiving a copy-and-paste operation, a selection of a block of code in an application, and a selection of a paste location in the application;

means for determining that the block of code has a size greater than a threshold;

means for creating a method that comprises the block of code; and

means for replacing the block of code with a first invocation of the method.

Hiew: See limitations addressed in claim 1. Col. 11: 39 – col. 12: 36, local computer accesses a remote computer...document manager window may be displayed to the user...displays the available files for the respective computers...The user may then select one or more of the available files for editing...perform various Windows command, such as, creating new

files...deleting...copying, cutting, and pasting...To open and edit a file displayed...drag a selected file from the remote panel...the file transfer program may communicate with the remote computer via the connectivity layer and instruct the remote computer to send the selected file ... Col. 13: 44-45, selected code is displayed on the editor window 82 the user can begin to edit the code Col. 13: 56-65, editor 80 works in connection with the parser layer 140...display a tree view 90 comprised of active elements 92 that are representative of code section that perform important functions...clicking on the desired active element 92, which causes the code that corresponds with that active element to be displayed in the code view panel 82b.

Hiew disclosed redirecting code, but failed to explicitly disclose adding a first invocation.

However, Hunt disclosed:

[0265] To apply inline redirection, the COIGN runtime, a collection of DLLs, is loaded into the application's address space before the application executes. One of these DLLs, the COIGN runtime executive (RTE), inserts the inline redirection code. [0263] Referring to FIG. 14, at load time, the first few instructions 502 of the target function 500 are replaced with a jump instruction 504 to the instrumentation detour function 506 in the instrumentation. The first few instructions 502 are normally part of the function prolog generated by a compiler and not the targets of any branches. The replaced instructions 502 are used to create a trampoline function 508. When the modified target function 501 is invoked, the jump instruction 504 transfers execution to the detour function 506 in the instrumentation. The detour function 506 passes

control to the remainder of the target function by invoking the trampoline function 508. After the moved instructions 502 are executed in the trampoline 508, a jump instruction 510 transfers execution back to a spot in the target function 501. The trampoline function 508 allows the detour function 506 to invoke the target function without interception.

Hiew / Hunt failed to suggest copying and pasting is done in response to a condition, test regarding the size of the block of code. (if the block of code has a size greater than the threshold)

However, Carini disclosed inlining code according to a cost function (col. 10: 23-28). The size of the procedure and the size of the region are given consideration prior to inlining (pasting in a method comprising a block of code).

Therefore, it would have been obvious, to one of ordinary skill in the art, to modify Hiew, using the teachings of Hunt, because Hiew recognized the need for (col. 1: 64-65) an Integrated Development Environment for generating and maintaining software code. Likewise Hunt recognized the benefit of [0014-0015] modifying an application, packaging it differently to optimize distribution and execution. It would have been obvious to further modify Hiew / Hunt using the teachings of Carini because Carini was also directed towards compiler optimization and site specific inlining. One would be motivated to place code into a location that was large enough to fit the code or otherwise locate the code elsewhere.

Per claim 9:

-means for adding a second invocation of the method at the paste location.

Hiew: Col. 14: 46-48, user can associate links or URLs with the template during creation (at paste location, link to URL

Per claim 10:

-means for selecting user interface options to present based on whether the copy-and-paste operation is within a single part and class.

See rejection of limitations addressed in claim 4 above.

Per claim 11:

-means for selecting user interface options to present based on whether the copy-and-paste operation is between two parts in a single package.

See rejection of limitations addressed in claim 5 above.

Per claim 12:

-means for selecting user interface options to present based on the copy-and-paste operation being between two parts in two different packages.

See rejection of limitations addressed in claim 5 above. Furthermore, Hiew (col. 11: 49-50):

User may then select one or more of the available files for editing...

Per claim 13:

A storage medium encoded with instructions, wherein the instructions when executed comprise:

receiving a copy-and-paste operation, a selection of a block of code in an application, and a selection of a paste location in the application;

determining that the block of code has a size greater than a threshold;

creating a method that comprises the block of code;

replacing the block of code with a first invocation of the method; and

adding a second invocation of the method at the paste location.

See rejection of limitations addressed in claims 8 & 9 above.

Per claim 14:

-selecting user interface options to present based on whether the copy-and-paste operation is within a single part and class.

See rejection of limitations addressed in claim 4 above.

Per claim 15:

-selecting user interface options to present based on whether the copy-and-paste operation is between two parts in a single package.

See rejection of limitations addressed in claim 5 above.

Per claim 16:

-selecting user interface options to present based on the copy-and-paste operation being between two parts in two different packages.

See rejection of limitations addressed in claim 12 above.

Per claim 17:

An electronic device comprising:

a processor; and a storage device encoded with instructions, wherein the instructions when executed on the processor comprise:

receiving a copy-and-paste operation, a selection of a block of code at a copy location in the application, and a selection of a paste location in the application;

determining that the block of code has a size greater than a threshold;

creating a method that comprises the block of code;

replacing the block of code at the copy location with a first invocation of the method;

adding a second invocation of the method at the paste location;  
determining that the block of code is to return a plurality of values; and  
creating a helper class for the method, wherein the helper class returns the plurality of values.

See rejection of limitations addressed in claims 1 & 8 above. Additionally, Hiew disclosed (col. 17: 14-16, 43-45) external files, library or macros that are referenced by the code capable of parsing and interpreting other file types such as SPSS, SQL, DB2, UDB, Oracle RDBMS, etc. (helper classes / return plurality of values).

Per claim 18:

-selecting a user interface default option to present based on whether the copy-and-paste operation is between two parts in a single package.

See rejection of limitations addressed in claim 5 above.

Per claim 19:

-designating a create private method option as a user interface default option if the copy-and-paste operation requested that the block of code be copied within a single part.

See rejection of limitations addressed in claim 4 above. Additionally, access modifiers, such as private, public, are well known in the art. Hiew disclosed a JAVA virtual machine (FIG. 2),

known to be object oriented, (col. 5: 34) routines, programs, objects, components, and/or data structures that perform particular tasks, (col. 11: 58-63) provided user interface default options. Col. 7: 15-22, The editor 80, the document manager 60 and the site manager 70 cooperate to track the association between edited code...The editor 80 may also employ language specific syntax checking (public / private access) and auto correct functions to enhance the software development capability of the integrated software application 30.

Per claim 20:

-selecting a user interface default option to present based on the copy-and-paste operation being between two parts in two different packages.

See rejection of limitations addressed in claim 12 above.

#### *Response to Arguments*

7. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

#### *Conclusion*

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

Application/Control Number:  
10/728,017  
Art Unit: 2191

Page 20

applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Mary Steelman/

**MARY STEELMAN**  
**PRIMARY EXAMINER**

Primary Examiner

02/07/2008